

An Improved ACO Algorithm for Multicast Routing

Ziqiang Wang and Dexian Zhang

School of Information Science and Engineering, Henan University of Technology,
Zheng Zhou 450052, China
wzqagent@xinhuanet.com

Abstract. The multicast routing problem with quality of service (QoS) constraints is a key requirement of computer networks supporting multimedia applications. In order to resolve QoS multicast routing effectively and efficiently, an improved ant colony optimization (ACO) algorithm is proposed to resolve this problem. The core idea of improved ACO algorithm is mainly realized through pheromone local and global updating rule. Experimental results show that this algorithm can find optimal solution quickly and has a good scalability.

1 Introduction

As a result of the emergence of many kinds of high-speed communication systems, such as ATM, and increasing demands of distributed multimedia applications, such as video on demand, multimedia conference, efficient and effective support of quality of service (QoS) has become more and more essential, these multimedia applications all require multicast support. Multicast employs a tree structure of the network to efficiently deliver the same data stream to a group of receivers. In multicast routing, one or more constraints must be applied to the entire tree. Multicast service is becoming a key requirement of computer networks supporting multimedia applications. In the past, the multicast routing problem has been formulated as the minimum cost multicast tree problem[1], i.e., the Steiner tree problem, which is well-known to be NP-complete[2]. Over the past decades, many works have done to solve multicast routing problems using conventional algorithm, such as exhaustive search routing and greedy routing. But due to the high degree of complexity, it is not practical to use these algorithms in real-time multicast routing. Recently, some nature-based algorithms[3] have been proposed. but these algorithms too complex and lower the efficiency of the algorithm.

Ant Colony Optimization (ACO) algorithm is a novel population-based meta-heuristic search algorithm[4] for solving difficult discrete optimization problems, inspired by the foraging behavior of real ant colonies. It has been applied to TSP, QAP, scheduling and graph coloring, etc. In this paper, we develop an efficient heuristic ant colony algorithm for multicast routing. This paper is organized as follows: The multicast routing model is described in section 2. The ACO algorithm are analyzed in section 3. The multicast routing algorithm based on ACO

algorithm are presented in section 4. Experimental results are given in section 5. Conclusions and future works are presented in section 6.

2 Multicast Routing Model

Communication network can be modeled as an undirected graph $G = \langle V, E \rangle$, where V is a finite set of vertices (network nodes) and E is the set of edges (network links) representing connection of these vertices. Each link in G has three weights $(B(x, y), D(x, y), C(x, y))$ associated with it, in which positive real values $B(x, y), D(x, y), C(x, y)$, denote the available bandwidth, the delay and the cost of the link respectively. Given a path $P(x, y)$ connected any two nodes x, y in G , it can be presumed that:

1) The delay of a path is the sum of the delays of the links $(a, b) \in P(x, y)$:

$$Delay(P(x, y)) = \sum D(a, b) \quad (1)$$

2) The available bandwidth of $(a, b) \in P(x, y)$ is considered as the bottle neck bandwidth of $P(x, y)$:

$$Width(P(x, y)) = \min(B(a, b)) \quad (2)$$

In Qos transmission of real time multimedia service, the optimal cost routing problem with delay and bandwidth constrained can be described as follows. Given $G = \langle V, E \rangle$, a source node s , and a multicast member set $M \subseteq V - \{s\}$, the problem is to find the multicast tree $T = (V_T, E_T)$ from source s to all destinations $v \in M$ and T must satisfy the following conditions:

$$Cost(T) = \min\left(\sum_{(x,y) \in (E_T)} C(x, y)\right) \quad (3)$$

$$\sum_{(x,y) \in (E_T)} D(x, y) \leq D_{max}, \forall v \in M \quad (4)$$

$$Width(P_T(s, v)) \geq W_{min}, \forall v \in M \quad (5)$$

where $P_T(s, v)$ is the set of links in the path from source nodes s to destination v in the multicast tree. Relation (3) means that the cost of multicast routing tree should be minimum. Relation (4) means that the delay requirement of Qos, in which D_{max} is the permitted maximum delay value of real time services. And relation (5) guarantees the bandwidth of communication traffic, in which W_{min} is the required minimum bandwidth of all applications.

3 The Ant Colony Optimization(ACO) Algorithm

3.1 The Ant System

The Ant Colony Optimization(ACO) technique[5] has emerged recently as a new meta-heuristic search methods for hard combinatorial optimization problems.

ACO algorithms have been inspired by the behavior of real ant colonies, in particular, by their foraging behavior. Real ants communicate with each other using an chemical substance called pheromone, which they leave on the paths they traverse. In the absence of pheromone trails ants more or less performance a random walk. However, as soon as they sense a pheromone trail on a path in their vicinity, they are likely to follow that path, thus reinforcing this rail. More specifically, if ants at some point sense more than one pheromone trail, they will choose one of these trails with a probability related to the strengths of the existing trails. This idea has first been applied to the TSP, where an ant located in a city choose the next city according to the strength of the artificial trails.

3.2 The ACO Algorithm Description

Informally, the ACO algorithm works as follows: m ants are randomly positioned on n cities. Each ant builds a tour path (i.e., a feasible solution to the TSP) by repeatedly applying a state transition rule. While constructing its tour, an ant also modifies the amount of pheromone on the visited edges by applying the local updating rule. Once all ants have finished their tours, the amount of pheromone on edges is modified again by applying the global updating rule. In the following, we firstly describe the ACO algorithm procedure, then we discuss the core steps of ACO algorithm: the state transition rule, the global updating rule, and the local updating rule.

Algorithm 1: The ACO Algorithm.

Initialize;

Loop;

Each ant is positioned on a starting node;

Loop;

Each ant applies a state transition rule to incrementally build a solution;

And a local pheromone updating rule ;

Until all ants have built a complete solution;

A global pheromone updating rule is applied ;

Until End condition.

3.3 State Transition Rule

While building a tour, ant k situated in city r moves to city s using the follow state transition rule:

$$s = \begin{cases} \arg \max_{u \in J_k(r)} \{[\tau(r, u)] \cdot [\eta(r, u)]^\beta\} : q \leq q_0 \\ S : & \text{otherwise} \end{cases} \quad (6)$$

where $J_k(r)$ is the set of cities not visited by ant k , $\tau(r, u)$ is the amount of pheromone trail on edge, $\eta(r, u)$ is a heuristic function which is the inverse of distance between city r and u , β is a parameter which determines the relative importance of pheromone versus distance, q is a random number uniformly distributed in $[0,1]$, q_0 is a parameter ($0 \leq q_0 \leq 1$), and S is a random variable

selected according to the probability distribution given by Equation (7) which gives the probability with which an ant k in city r choose the city s to move to:

$$p_k(r, s) = \begin{cases} \frac{[\tau(r, u)] \cdot [\eta(r, u)]^\beta}{\sum_{u \in J_k(r)} \tau(r, u) \cdot [\eta(r, u)]^\beta} : S \in J_k(r) \\ 0 : & otherwise \end{cases} \quad (7)$$

3.4 Local Updating Rule

While building a solution of the TSP, ants visit edges and change their amount of pheromone trail by applying the local updating rule of Equation (8):

$$\tau(r, s) \leftarrow (1 - \rho) \cdot \tau(r, s) + \rho \cdot \tau_0 \quad (8)$$

where ρ ($0 < \rho < 1$) is the pheromone decay parameter, τ_0 is an initial pheromone level.

3.5 Global Updating Rule

Global updating rule is performed after all ants have completed their tours. The pheromone level is updated by applying global updating rule of Equation(9).

$$\tau(r, s) \leftarrow (1 - \alpha) \cdot \tau(r, s) + \alpha \cdot \Delta_k \tau(r, s) \quad (9)$$

where α ($0 < \alpha < 1$) is the pheromone decay parameter,

$$\Delta_k \tau(r, s) = \begin{cases} (L_{gb})^{-1} : (r, s) \in global_best_tour \\ 0 : & otherwise \end{cases} \quad (10)$$

and L_{gb} is the length of global best tour from the beginning of the trail. Equation (9) indicates that only those edges that belong to the global best tour will receive reinforcement.

4 The ACO Algorithm For Multicast Routing

The proposed ACO algorithm steps for multicast routing are as follows:

- Step 1:* Initialize network nodes;
- Set $NC:=0$; $\tau_0:=c$;
- Put m ants to the source node;
- Step 2:* Check bandwidth of all edges, deletes those edges that do not satisfy the bandwidth require;
- Step 3:* Setup tabu table;
- Set $r:=1$;
- For $k:=1$ to m
- Put the value of source node into tabu table $tab_k(r)$;
- Step 4:* Repeat this step until the tabu table $tab_k(r)$ is full;

Set $r:=r+1$;
 For $k:=1$ to m
 Random choose q and compare with q_0 ;
 If $q \leq q_0$
 Then choose the next node s according to

$$s := \arg \max_{u \in J_k(r)} \{[\tau(r, u)] \cdot [\eta(r, u)]^\beta\} \quad (11)$$

Else choose the node s according to the probability distribution given by Equation (7).

Compute the delay to reach node s , and compare the result with delay D_{max} ;
 If the result exceeds the constraints D_{max}
 Then the k th ant choose a new node;
 Else
 {
 The k th ant moves to node s , and put s into $tab_k(r)$;
 Pheromone local updating according to

$$\tau(r, s) := (1 - \rho) \cdot \tau(r, s) + \rho \cdot \tau_0 \quad (12)$$

}
 Step 5: Compute $\Delta_k \tau(r, s)$ and pheromone global updating;
 For $k:=1$ to m set

$$\Delta_k \tau(r, s) := \begin{cases} (L_{gb})^{-1} & : (r, s) \in global_best_tour \\ 0 & : otherwise \end{cases} \quad (13)$$

$$\tau(r, s) := (1 - \alpha) \cdot \tau(r, s) + \alpha \cdot \Delta_k \tau(r, s) \quad (14)$$

$NC:=NC+1$;
 Step6: Check stop condition;
 If ($NC < NC_{max}$)
 Then empty all tabu table, and goto Step3;
 Else Print the minimum cost multicast tree.

5 Experimental Results

We have performed simulation to investigate the performances of multicast routing algorithms based on ACO algorithm. A random generator developed by Salama[2] is used to create links interconnecting the nodes. The random graphs are generated using the above graph generator with an average degree 4, which have the appearance roughly resembling that of geographical maps of major nodes in the Internet. The source and the destination are randomly generated. D_t for destination t is uniformly distributed in range [30ms,160ms]. The bandwidth and delay of each link are uniformly distributed in range [10,50] and [0,50ms] respectively. The cost of each link is uniformly distributed in range [0,200]. In

Table 1. CPU Time of ACO Algorithm

Nodes	Edges	Time(s)
20	32	0.27
40	89	0.45
80	172	1.42
120	239	2.68
160	336	5.37
180	371	8.19
200	427	9.73

Table 2. Predictive Accuracy Comparison

Algorithm	Optimal Solutions	Sub-optimal Solutions	Invalid Solutions
HNN	75.6%	21.5%	2.9%
GA	78.4%	19.4%	2.2%
IA	78.9%	19.6%	1.5%
ACO	79.9%	18.2%	1.9%

Table 1, we have compared the CPU times of ACO algorithm for different combinations of node and edge.

Table1 results clearly show that the running time of ACO algorithm grows very slowly with the size of the network. Therefore, our algorithm is very effective.

Furthermore, for the same multicast routing, we made 300 simulations by ACO algorithm against Hopfield Neural Networks(HNN)[6],GA[7] and immune algorithm(IA)[8]. The computation results are shown in Table 2.We can find that ACO algorithm performances better than HNN,GA and IA. So our proposed algorithm has good performance.

6 Conclusions

Multicast routing arises in many multimedia communication applications, and this problem has been proved to be NP-Complete. In this paper, we studied the bandwidth-delay-constrained least-cost multicast routing problem, and present an ACO algorithm to solve the problem. The simulation results show that this algorithm is an efficient algorithm. For further research, we will improve the effectiveness of ACO algorithm and study the multicast routing problem with general constraint.

References

1. Wang,Z.,Crowcroft,J.:Quality of service for supporting multimedia applications.IEEE Journal on Selected Areas in Communications.14(1996)1228–1234

2. Salama,H.F.,Reeves,D.S.,Viniotis, Y.:Evaluation of multicast routing algorithms for real-time communication on high-speed networks. *IEEE Journal on Selected Areas in Communications*. 15(1997)332–345
3. Wang,Z.,Shi,B.:Solution to Qos multicast routing problem based on heuristic genetic algorithm. *Journal of Computer*.24(2001)55–61
4. Dorigo,M.,Maniezzo,V. and Colomi A.:The ant system: optimization by a colony of cooperation agents. *IEEE Transaction on System, Man, and Cybernetics-Part B*. 26(1996)1–13
5. Dorigo,M.,Gambardella, L.M.: Ant colony system:a cooperative learning approach to the traveling salesman problem.*IEEE Transaction on Evolutionary Computation*.1(1997)53–66
6. Chotipat,P.,Goutam,C.,Norio S.:Neural network approach to multicast routing algorithms for real-time communication on high-speed networks.*IEEE Journal on Selected Areas in Communications*.15(1997)332–345
7. Feng,X.,Li,J.Z.,Wang J.V.,et al.:Qos routing based on genetic algorithm. *Computer communication*.22(1999)1392–1399
8. Liu,F.,Feng,X.J.:Immune algorithm for multicast routing.*Chinese Journal of Computer*.26(2003)676–681